

### **Remarks**

This Reply is in response to the Final Office Action mailed December 30, 2009.

#### **I. Summary of Examiner's Rejections**

Prior to the Office Action mailed December 30, 2009, Claims 11, 19 and 23-30 were pending in the Application. In the Office Action, Claims 11, 19 and 23-30 were rejected as being unpatentable over Carey (U.S. Patent No. 6,947,945) in view of Leong (U.S. Patent No. 7,020,641).

#### **II. Summary of Applicants' Amendments**

The present Response does not amend any claims. Reconsideration of the Application in light of the following remarks is respectfully requested.

#### **III. Claim Rejections under 35 U.S.C. § 103(a)**

In the Office Action mailed December 30, 2009, Claims 11, 19 and 23-30 were rejected as being unpatentable over Carey (U.S. Patent No. 6,947,945) in view of Leong (U.S. Patent No. 7,020,641).

#### **Claim 11**

Claim 11 currently defines the following features:

11. *A computer-implemented method comprising:  
generating a transformation file by employing a query language, said transformation file containing a set of rules to transform data between two or more formats having different shapes;  
attaching the transformation file to a workflow, such that the set of rules are referenced from inside the workflow;  
associating, at compile time, a first shape of a first data structure with an intermediate shape representation based on the set of rules of the transformation file, wherein the first shape defines a structure and layout of data in the first data structure;  
receiving a second data structure during runtime execution of said workflow, said second data structure having a second shape that is different from the first shape of the first data structure;  
applying the intermediate shape representation to the second data structure;  
mapping the second data structure from the intermediate shape representation to the first shape of the first data structure; and  
generating a runtime object containing the data obtained from the second data structure and having the first shape of the first data structure and using the runtime object as input for a component of said workflow.*

As such, Claim 11 defines generating a transformation file and attaching it to a workflow, such as a business process management (BPM) workflow. Once the file is attached to the workflow, the rules of the transformation file are referenced from inside the workflow.

In general, BPM workflow systems are used to automate processes within an enterprise. The workflow is a software component that performs a specific series of tasks that are connected in a way that allows them to be ordered according to the completion of tasks. In a workflow, information such as files, documents or tasks are passed as XML messages. However, in order for the BPM workflow to invoke other systems (e.g. J2EE component), there must be a way to transform from XML language to the language of those components (Background, paragraphs 8-9).

By attaching a transformation file to the workflow, the rules specified therein, can be accessed from inside the workflow. Thus, at compile time, an intermediate shape representation is associated with a particular shape of a data structure according to the set of rules of the transformation file. Subsequently, during runtime execution of the workflow, data structures may be received which have a different shape. At this point, the intermediate shape representation is applied to generate a runtime object from the different shape data structure, where the runtime object is then in the right shape to be used as input for a component of the workflow (e.g. a particular task). In this manner, the transformation file that is attached to the workflow can be used to transform between different data shapes in order to allow the workflow to be easily integrated with systems that use data in different formats and shapes.

Carey teaches using an XML query language to publish relational data as XML. In particular, Carey appears to describe a techniques for translating XML queries into queries against relational databases. "A server uses a relational database as its data store. A mapping is established from each table in the database to a virtual XML document. Clients (or other servers) query these virtual documents using XML-QL. An XML-Query is transformed into a language-neutral intermediate representation. The intermediate representation is then translated into an SQL query over the underlying relational table and into tagging instructions..." (Carey, col. 3, lines 10-25).

Leyong teaches techniques for maintaining a database of data objects. In particular, the cited portions of Leyong disclose logic implemented in the database interface to handle a request from non-Java or Java application to add a non-Java or Java data object to an object oriented database. If the received object is a Java object, then the interface calls an OOD API to add the data object directly to the database. Otherwise, if the received object is a non-Java data object that is received along with an XML-schema file, then the class name and attribute name, data type and

length information included in the XML schema file are loaded into memory as one class information set. The interface then creates a Java source file to code the class defined in the XML schema. (Leyong, col. 5, lines 45-67).

However, Applicant respectfully submits that Carey in combination with Leyong fail to disclose or render obvious the features of Claim 11.

In particular, Carey and Leyong fail to disclose the features of attaching the transformation file to a workflow, such that the set of rules are referenced from inside the workflow, receiving a second data structure during execution of the workflow, translating that second structure into a runtime object by using the transformation file and using the runtime object as input for a component of said workflow, as defined in Claim 11. The concept of workflows or BPM systems is not at all disclosed in the cited references. More importantly, there is no transformation file attached to any workflow and there is no transforming of objects that are passed between the workflow components, as defined in Claim 11. Both references are silent regarding this feature.

At a high level, both Carey and Leyong appear to be concerned with accessing a database in some manner. For example, Carey describes a method for translating an XML-QL query into a standard SQL query by maintaining server-side virtual tables of the actual database tables. Thus, Carey's translation techniques appear to be invoked when a client accesses a database. Similarly, Leyong describe a method for adding data to a database by reading an XML schema that comes along with a non-Java object so that the information can be added appropriately to the database. In contrast, the features of Claim 11 are not concerned with accessing a database but are instead invoked between runtime tasks of a workflow. For example, in Claim 11, the transformation file is attached to the workflow and during compile time, the intermediate representation is associated with the data shape of the first data structure so that when data structures of other shapes are received during execution of the workflow, that representation can be used to create appropriate runtime data objects that are used as input for the workflow next task.

In addition, Carey and Leyong fail to disclose the feature of associating an intermediate representation with *a first shape of a first data structure*, as defined in Claim 11. Rather, the cited reference Carey instead discloses that a parser "converts an XML query to a language neutral intermediate representation." (col. 4, lines 22-26). Thus, Carey describes a technique for creating an abstracted representation of a query, not an intermediate representation of a *data structure shape*, as defined in Claim 11. Translating a query from one language into another, is not the same as creating an intermediate representation of a data *shape*, which is later used at runtime. In fact,

Claim 11 is not concerned with querying the database for information, but instead uses the representation to translate between data objects of different shapes at runtime.

In view of the above comments, Applicants respectfully submit that Claim 11, as amended, is neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

#### **Claims 19 and 26**

Claims 19 and 26, while independently patentable, recite limitations that, similarly to those described above with respect to Claim 11, are not taught, suggested nor otherwise rendered obvious by the cited references. Reconsideration thereof is respectfully requested.

#### **Claims 23-25 and 27-30**

Claims 23-25 and 27-30 are not addressed separately, but it is respectfully submitted that these claims are allowable as depending from an allowable independent claim, and further in view of the comments provided above. Applicants respectfully submit that Claims 23-25 and 27-30 are similarly neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

It is also submitted that these claims also add their own limitations which render them patentable in their own right. Applicants respectfully reserve the right to argue these limitations should it become necessary in the future.

#### **V. Conclusion**

In view of the above amendments and remarks set forth above, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and reconsideration thereof is respectfully requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Application No.: 10/784,376  
Office Action dated: December 30, 2009  
Reply dated: January 30, 2010

Date: March 1, 2010

By: /Justas Geringson/  
Justas Geringson  
Reg. No. 57,033

Customer No. 80548  
FLIESLER MEYER LLP  
650 California Street, 14<sup>th</sup> Floor  
San Francisco, California 94108  
Telephone: (415) 362-3800